import pandas as pd

url="https://raw.githubusercontent.com/SP20-BCS-126/IDS/Assignment-4/gender-prediction.csv"

df=pd.read\_csv(url)

display(df)

#install scikitplot

!pip install scikit-plot

#import libraries

from sklearn import preprocessing

import pandas as pd

#import different ML classifiers

from sklearn.naive\_bayes import GaussianNB, BernoulliNB, MultinomialNB

from sklearn.svm import SVC, LinearSVC

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.ensemble import RandomForestClassifier

from sklearn.neural\_network import MLPClassifier

#import ML evaluation metrics

from sklearn.metrics import accuracy\_score, f1\_score

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics, model\_selection

#import scikitplot to plot confusion matrix

import scikitplot as skplt

gender=df['gender']

height=df['height']

weight=df['weight']

beard=df['beard']

hairlength=df['hair\_length']

shoesize=df['shoe\_size']

scarf=df['scarf']

eyecolor=df['eye\_color']

labels = preprocessing.LabelEncoder()

g\_encoded = labels.fit\_transform(gender)

b\_encoded = labels.fit\_transform(beard)

hl\_encoded = labels.fit\_transform(hairlength)

s\_encoded = labels.fit\_transform(scarf)

e\_encoded = labels.fit\_transform(eyecolor)

x = list(zip(height,weight,b\_encoded,hl\_encoded,shoesize,s\_encoded,e\_encoded))

y = g\_encoded

X\_train, x\_test, Y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.33, random\_state = 2)

#model = RandomForestClassifier()

#model = SVC()

model = MLPClassifier()

model.fit(X\_train,Y\_train)

prediction = model.predict(x\_test)

model\_acc = accuracy\_score(y\_test, prediction)\*100

print(model\_acc)

model\_cl\_rep = metrics.classification\_report(y\_test, prediction)

print(model\_cl\_rep)

X\_train, x\_test, Y\_train, y\_test = train\_test\_split(x, y, test\_size = 0.2, random\_state = 2)

model.fit(X\_train,Y\_train)

prediction = model.predict(x\_test)

model\_acc = accuracy\_score(y\_test, prediction)\*100

print(model\_acc)

model\_cl\_rep = metrics.classification\_report(y\_test, prediction)

print(model\_cl\_rep)

x = list(zip(height,weight,shoesize,s\_encoded,e\_encoded))

y = g\_encoded

#and then to segment 35

model\_cm = metrics.confusion\_matrix(y\_test, prediction)

print(model\_cm)

#Question4

model = GaussianNB()

model.fit(x, y)

GaussianNB()

new\_input = [(65, 120, 0, 3, 36, 1, 0), (68, 180, 1, 0, 40, 1, 2), (72, 180, 1, 2, 44, 0, 1), (67, 170, 0, 0, 40, 1, 1),(70, 176, 0, 3, 44, 1, 1)]

new\_output = ['female','male','male','female','female']

no\_encoded = labels.fit\_transform(new\_output)

prediction = model.predict(new\_input)

model\_acc = accuracy\_score(no\_encoded, prediction)\*100

print(model\_acc)

model\_cl\_rep = metrics.classification\_report(no\_encoded, prediction)

print(model\_cl\_rep)

model\_cm = metrics.confusion\_matrix(no\_encoded, prediction)

print(model\_cm)